

Řízení kvality v procesu implementace informačního systému

Každá organizace, která implementuje nebo rozšiřuje svůj informační systém, a o to více každý dodavatel implementace, na otázku, zda provádí řízení kvality, jednoznačně odpoví, že ano. Pokud se však začneme ptát, jakými konkrétními činnostmi je toto prováděno, odpověď již tak jednoznačná často nebývá. V tomto článku se tedy zaměříme na to, jak by se řízení kvality (angl. Quality Management) v projektu implementace informačního systému mělo provádět.

Definice pojmů

Řada nedorozumění a chyb v této oblasti plyne již z toho, že není jednoznačně stanoveno, co pojmem **kvalita** (někdy česky též jakost) rozumíme. Co je tedy kvalita? Moje dcera, když jí byly čtyři roky, si jednou neopatrně hrála se svou hračkou. Když to viděla její teta, upozornila ji, že se jí hračka rozbije. Na to dcera odpověděla: „Nerozbije, to je kvalitní“. Ano, v běžném životě vystačíme s intuitivním a tedy i subjektivním chápáním kvality. Při **řízení** kvality však s tímto přístupem nevystačíme. Podíváme-li se do projektových metodik na definici pojmu kvalita, zjistíme, že to ne první pohled není moc srozumitelné. Dle PMBOK (podklad pro PMP certifikaci) je to: „the degree to which a set of inherent characteristics fulfill requirements“. Podle PRINCE2 je to: „product's ability to meet its requirements“. Dle ČSN EN ISO 9000:2000 je kvalita definována, jako je stupeň splnění požadavků souborem inherentních znaků.

Co to tedy vlastně znamená? Převáděno do běžné mluvy jde tedy o to, aby výstupy projektu implementace (nejenom vlastní software!) splňovaly požadavky, které na ně jsou kladeny. V žádném případě se tedy nejedná o vyhovění všem požadavkům a přáním zákazníka, množství funkcionalit, nejedná se o to, jak je systém moderní a luxusní, a dokonce ani o to, že systém je prostý jakékoli vady či odolný vůči destrukci.

Pro řízení kvality implementace IS je tedy důležité stanovit jasně, přesně a jednoznačně požadavky na vlastnosti a způsob, jakým se bude ověřovat jejich naplnění. To samo o sobě není jednoduchou činností, protože velmi často jsou požadavky a potřeby cílového zákazníka definovány velice vágně.

Jako v každém jiném projektu, i při implementaci informačního systému můžeme činnosti související s řízením kvality podle obou výše zmíněných metodik rozdělit do tří oblastí:

1. Vytvoření plánu řízení kvality
2. Kontrola kvality
3. Zajištění kvality

Plán řízení kvality stanovuje prvky, zásady a skladbu činností a záznamů nezbytných k úspěšnosti projektu z hlediska kvality. Opět převáděno do běžné mluvy: co, kdo, jak a kdy bude ohledně řízení kvality dělat.

Kontrola kvality (angl. Quality Control) zahrnuje sledování konkrétních výstupů projektu ke zjištění, zda jsou v souladu s příslušnými požadavky a určení způsobů, jak eliminovat příčiny neuspokojivých

výstupů. Do běžné mluvy: jde o to, aby to, co implementace vyprodukuje (vlastní software, dokumentace atd.) mělo požadované vlastnosti.

Zajištění kvality (angl. Quality Assurance) je aplikace plánovaných, systematických činností, které zajistí, že projekt využívá všech procesů potřebných ke splnění požadavků na kvalitu. Záruku kvality obvykle provádí jednotlivec nebo skupina, která se aktivně nepodílí na realizaci projektu. Do běžné mluvy: jedná o kvalitu vlastních projektových činností.

Jednu věc je třeba zdůraznit, protože bez ní není možno pochopit celý koncept kvality v řízení projektu: rozlišuje se řízení kvality výstupů, tj. Kontrola kvality (např. software, uživatelské příručky, data v systému,...) a řízením kvality toho, jak probíhají projektové práce, tj. Zajištění kvality (např. řízení harmonogramu, školení koncových uživatelů, vytváření uživatelských příruček,...).

Konflikt kvality a nákladů

Ve své praxi jsem se opakovaně setkal s námitkou, že provádět všechny činnosti řízení kvality tak, jak je předepisuje metodika, neúměrně prodražuje implementační projekt. Na straně dodavatele to dokonce povede k tomu, že jeho cena bude nekonkurenceschopná. Kvalita samozřejmě něco stojí. Jde ale o to, aby náklady vynaložené na řízení kvality byly vyváženy odpovídajícími přínosy.

Rozsah činností managementu kvality musí odpovídat rozsahu projektu a důležitosti výsledného softwarového řešení. Pro velký program implementace komplexního informačního systému, sestávajícího z několika softwarových produktů a s tisícičkami člověkodů práce je užitečné mít samostatný Plán řízení kvality, externě prováděný audit kvality, procedury zlepšování procesů atd.

Naproti tomu pro malý projekt implementovaný malým týmem zkušených odborníků místo samostatného dokumentu plně postačuje několik odstavců v Plánu projektu (nutno rozlišit „Plán“ a „Harmonogram“) s odkazem na zvyklosti v rámci organizace a specifikováním odlišností pro konkrétní projekt. Místo externího auditu lze použít checklist z metodiky, zlepšování procesů nemá smysl atd.

I pro řízení kvality totiž platí věta z PMI metodiky, kterou já osobně považuji za jednu z nejdůležitějších pro projektové řízení: „Scope Management obsahuje procesy potřebné k zajištění toho, že projekt zahrnuje všechny potřebné činnosti a pouze činnosti potřebné k jeho dokončení“. Jinými slovy, pokud nějaké činnosti řízení kvality nejsou pro dosažení cílů konkrétního projektu nezbytné, nebudou se provádět. To však neznamená, že v jiném projektu je možno tyto činnosti též vynechat a navíc je třeba toto vynechání udělat jako vědomé rozhodnutí se znalostí důsledků.

Je potřeba též připomenout, že mnoho z aktivit, o kterých v tomto článku hovoříme, se na většině implementačních projektů nějakým způsobem tak jako tak provádí. Řízení kvality tedy někdy znamená pouze zavést strukturovaný přístup k těmto aktivitám a řídit je s vědomím, že jejich účelem je zajistit kvalitu výstupů projektu.

Ke snížení nákladů na řízení kvality může výrazně přispět, pokud jsou k dispozici dobře připravené šablony a popisy procedur, které se použijí opakovaně, a vyškolený (ideálně seniorní) tým. To by mělo být zodpovědností PMO. Pro malé organizace se předpokládá, že toto dodá dodavatel implementace.

Také je nutno provést porovnání nákladů na dosažení kvality versus nákladů na nekvalitu. Ačkoli řada učebnic projektového managementu tvrdí, že náklady na odstranění vady rostou s dobou, která uplynula od zahájení projektu, při započtení nákladů na kontrolu kvality tomu tak vždy být nemusí. V praxi jsem se například setkal se situací, kdy management klienta určil skupinu svých odběratelů, pro které musely být dokumenty vytištěny vždy správně a u těch bylo nutno provést důkladné zkoušky a kontroly. Pro ostatní jeho odběratele stačilo provést pouze namátkové zkoušky, protože dopad případně chybně vytištěných dokumentů na byznys klienta by velmi malý oproti nákladům na testování a kontrolu. Každé takovéto rozhodnutí je však nutno provést po úvaze.

Plán řízení kvality

Plán řízení kvality popisuje (případně odkazuje se na) požadavky na kvalitu výsledného produktu, způsob, jakým se naplnění těchto požadavků bude ověřovat. Dále činnosti a procedury v řízení kvality včetně potřeb zdrojů a to jak pro kontrolu kvality, tak pro zajištění kvality. Podrobněji se těmto oblastem budeme věnovat v následujících kapitolách. Pro plánování kontroly kvality připomeňme základní zásadu moderního řízení kvality: kvalita je plánována, dizajnována a je integrální součástí činností – není to revize.

Klíčovou součástí Plánu řízení kvality je při implementaci IS **Plán testování**. Měl by být sestaven již v úvodní fázi projektu. Testování je samostatná a složitá disciplína a v tomto článku se jí z důvodu rozsahu nebudeme podrobně věnovat. Řekněme si pouze, že Plán testování by měl obsahovat následující kapitoly:

1. Cíle a požadavky testování v projektu
2. Předpoklady
3. Rozsah testování
4. Typy testování
5. Přístup k testování
6. Nástroje pro provádění a řízení testů
7. Pokrytí byznys požadavků
8. Kritéria úspěšnosti testů
9. Testovací prostředí
10. Defekt Management
11. Role a zodpovědnosti

Protože úspěšné testování bývá jednou z klíčových podmínek pro předání projektu, je důležité, aby Plán testování byl schválen zákazníkem.

Kontrola kvality

Uveďme si několik příkladů, jak se kontrola kvality provádí při implementaci IS. Příklady vycházejí z klasické implementace typu waterfall, ale obdobně se dají aplikovat i na agilní projekty.

Jedním z nejdůležitějších výstupů implementačního projektu je popis požadavků na fungování informačního systému. Často se označuje jako koncept a bývá připravován ve spolupráci odborníků na implementaci s vlastníky procesů a klíčovými uživateli. Obvykle podléhá závěrečnému schválení, což samo o sobě je aktivita kontroly kvality. Problém ale může být, že dokument se reviduje až po

jeho dokončení, což znamená, že kontrola kvality je zajišťována revizí. Ve svém důsledku to může v nepříznivých případech vést k požadavkům na zásadní přepracování několika kapitol, či případně celého dokumentu.

Mně se v praxi osvědčilo začít jednou vzorovou oblastí (procesu, funkcionality,...), její popis nechat schválit (pokud možno pracovníky, kteří se nejsou jejími přímými autory) a poté ji použít jako „šablonu“ pro další kapitoly. Další kapitoly pak schvalovat průběžně, jak jsou vypracovávány. Celková revize dokumentu na závěr je pak již poměrně rychlou záležitostí. Výhoda takového postupu spočívá (kromě jednotného zpracování dokumentace) v tom, že pokud ze strany implementačního týmu došlo k nějakému zásadnímu nepochopení potřeb zákazníka nebo k chybě, která se může objevovat ve více kapitolách, je tento nedostatek odhalen relativně brzy a není nutno přepracovávat velkou část dokumentu.

Obdobné zásady, jako pro výše popsany seznam požadavků resp. koncept, se dají použít pro vypracování prakticky jakéhokoli rozsáhlejšího dokumentu resp. souboru dokumentů stejného typu, jako třeba manuály, školicí dokumentace a dokonce i testovací skripty.

K testování pouze poznamenejme, že i ono by mělo naplňovat princip integrální a nikoli revidované kvality. To znamená testovat v průběhu vývoje malé samostatné části softwaru nejdříve implementačním týmem a poté klíčovými uživateli zákazníka. Například otestovat založení objednávky, i když k ní ještě není možno zaúčtovat fakturu. Nebo pořídit příjem na sklad, i když systém ještě špatně počítá účetní hodnotu zboží. Ve scénářových a integračních testech se testuje pouze návaznost jednotlivých činností a integrace. Je zásadní selhání projektového týmu, pokud se až v integračních testech zjistí, že například systém netiskne správně objednávku. Výhodou popsaného postupu opět je, že vady jsou odhaleny relativně brzy a je tedy čas je opravit. Minimalizuje se tak riziko, že při nalezení velkého množství vad v integračních testech je nutno posunout milník projektu.

Ještě připomeňme, že je potřeba otestovat všechny požadované vlastnosti systému. Tedy kromě aplikační funkcionality též schopnost zpracovat požadované množství dat v požadovaném čase, zálohování a reload zálohy, tisky na všech typech tiskáren, autorizace atd.

Další samostatnou kapitolou jsou data (kmenová i transakční), která budou v rámci projektu do systému nahrávána. Zde je třeba kontrolovat kvalitu jak programů pro jejich nahrání (stejným způsobem jako testy aplikací – viz výše), tak správnost a úplnost vlastních dat.

Výčet v této kapitole není úplný a další příklady by bylo možno doplnit dle typu projektu.

Zajištění kvality

Jak jsme se již zmínili výše, zajištění kvality (Quality Assurance) má za úkol ověřit kvalitu provádění projektových činností a měla by být prováděna někým, kdo stojí mimo projekt. Může to být jiné oddělení z organizace koncového zákazníka (Oddělení jakosti, PMO,...) nebo třetí strana nebo kombinace obojího. Implementační partner obvykle provádí též zajištění kvality, ale to nelze považovat za plnohodnotné zajištění kvality implementačního projektu, protože cíle implementačního partnera nejsou zcela totožné s cíli organizace, pro kterou se implementace provádí.

V každém případě je provádění zajištění kvality vysoce odbornou a seniorní záležitostí. Pokud se provádí pouze formální kontrola projektových dokumentů juniorními pracovníky podle obecných návodů, nemá to pro projekt přidanou hodnotu a je to pouze ztrátou času pro projektový tým.

Zajištění kvality se obvykle provádí před koncem jednotlivých fází projektu, typicky po úvodním plánování, před schválením konceptu, před integračními testy, před ukončením vytváření řešení (angl. solution built), před zahájením přechodu do produktivního provozu a před ukončením projektu. Kontinuální zajištění kvality stálou přítomností na projektu je příliš nákladné a místo vlastního zajištění kvality plní spíše funkci koučingu projektového vedení.

Zajištění kvality je v zásadě možno provádět dvěma způsoby:

Revizí projektových dokumentů: Provádějící osoba (auditor) zkoumá úplnost a kvalitu projektové dokumentace, zda jsou vypracovány všechny dokumenty, které jsou pro daný stav projektu relevantní, zda jsou spravované a aktuální. Nejedná se o mechanickou kontrolu. Auditor se musí umět v dokumentaci zorientovat, pochopit, jaká dokumentace a v jaké podrobnosti je pro daný typ projektu důležitá, vzít v úvahu, že dokumenty mohou mít nejenom jiné názvy, ale že informace mohou být mezi dokumenty rozděleny různým způsobem, musí umět posoudit, které chyby v dokumentaci jsou pro kvalitu projektu zásadní a které jsou pouze formální atd. Pokud možno též odhadnout, zda předložený dokument je opravdu živým dokumentem nebo zda se jedná pouze o formální splnění požadavků metodiky (např.: Znamená předložený registr rizik, že rizika jsou na projektu pravidelně spravována nebo se jedná pouze o nějak vyplněnou tabulku, aby se splnil požadavek metodiky?).

Tato forma zajištění kvality v sobě skrývá riziko špatného pochopení jejích výsledků. Ze své praxe znám případ, kdy výsledný report říkal, že kontrola nenašla žádnou závadu a přesto se jednalo o vysoce problémový projekt, který již v té době byl ve zpoždění a s přečerpaným rozpočtem. Jak je to možné? Tato kontrola totiž neříká nic o stavu projektu, pouze o stavu projektové dokumentace. V tomto případě projektová dokumentace jasně říkala, že projekt má zásadní trable a managementu organizace byla tato informace dobře známa.

Pokud se však jedná sice o neschopné projektové vedení, ale schopné falzifikátory (jakkoli pro projektového manažera je to velmi krátkozraká strategie), nemusí v tomto případě kontrola problémový projekt odhalit.

Review projektového vedení: Kromě kontroly projektových dokumentů se provádí ještě pohovory s různými podílníky (angl. stakeholdery) projektu. Dle aktuální projektové fáze to jsou např.: koncový uživatel, programátor, klíčový uživatel, konzultant dodavatele, správce systému, projektový manager, sponzor projektu atd. Tento způsob je náročnější na čas projektového týmu i auditora a tedy nákladnější.

Zároveň je velmi náročný na znalosti a zkušenosti auditora. Auditor musí vhodně zvolenými otázkami a z obdržených odpovědí poznat skutečný stav projektu a odhalit příslušná rizika pro úspěch projektu. Jako auditor jsem se například setkal se situací, kdy mi vedení projektu předložilo výborný harmonogram, ale několik klíčových pracovníků projektu mi nedokázalo odpovědět na „základní“ otázku, na kdy je plánován začátek produktivního provozu. To pro mě byl signál nejen o chybách v řízení harmonogramu, ale i o chybách v řízení komunikace na projektu.

Součástí závěrečné zprávy by měl být seznam konkrétních nalezených nedostatků, z nich vyplývajících rizik (včetně ohodnocení z hlediska pravděpodobnosti a dopadu rizika) a konkrétní doporučené činnosti pro jejich odstranění nebo zmírnění. Zajištění kvality však nemá sloužit jako nástroj k obviňování v projektovém týmu nebo jako zbraň při boji zákazníka s dodavatelem implementace (případně naopak). Je to prostředek k zvýšení pravděpodobnosti úspěchu projektu. A mělo by se brát vážně. Ve své praxi jsem bohužel několikrát slyšel: „nojo, vy jste nám to do té zprávy napsal, že to takhle může dopadnout“.

Externí zajištění kvality popsané v této kapitole se může jevit jako zbytečný náklad projektu. Ale kolik stojí neúspěšný projekt, překročení harmonogramu, nákladů, nedostatečná funkčnost systému, chyby v datech,...? Najímáte si stavební dozor na stavbu, auditory na kontrolu účetnictví, finanční poradce při akvizicích,...? A to nejsou to zbytečné náklady?